

BACKEND DOCUMENTATION:

Our backend API is developed on RUBY ON RAILS web framework.

Our app uses:

Rails version — Rails 7.0.4

Ruby version — ruby 2.7.1

Github Url: https://github.com/divyansh-23/finadvice_app

Our App's backend API is deployed On Heroku: Heroku app link:

<https://radiant-stream-07739.herokuapp.com/>

Our App's Live URL: <https://silly-frangipane-f9d81d.netlify.app>

Database Used: POSTGRESQL

Here are some of the curl requests we used while developing and testing the backend API.

For SESSIONS:-

Registrations or Signup or User Creation:

```
curl --header "Content-Type: application/json" --request POST --data '{"user": {"email": "12divyansh1232@dev.com", "password": "asdfasdf", "password_confirmation": "asdfasdf", "name": "Divyansh Shrivastava", "phone": "2222222222", "role": "instructor", "dob": "2000/01/22"}}' http://localhost:3000/registrations
```

Login —

```
curl --header "Content-Type: application/json" --request POST --data '{"user": {"email": "divyansh@dev.com", "password": "asdfasdf"}}' http://localhost:3000/sessions
```

Response:

```
{ "status": "created", "logged_in": true, "user": { "id": 1, "email": "divyansh@dev.com", "password_digest": "$2a$12$HMyZE62gUkYrvV0.6swGeMO5GiG5ulf0jhbaot9aMyjbJvaGd/7i", "created_at": "2022-11-28T18:59:37.014Z", "updated_at": "2022-11-28T18:59:37.014Z", "role": null, "dob": null, "phone": null, "name": null }}
```

For COURSES:

Create –

```
curl --header "Content-Type: application/json" --request POST --data '{"course": {"name": "Data Structure and Algorithms", "subject": "computer science", "category": "beginners", "description": "A complete introduction to data structures and algorithms", "start_date": "Tue, 29 Nov 2022", "course_enroll_limit": 100, "user_id": 5}}' http://localhost:3000/courses
```

Command to create a course on rails console.

```
Course.create!( "name": "Data Structure and Algorithms", "subject": "computer science", "category": "beginners", "description": "A complete introduction to data structures and algorithms", "start_date": "Tue, 29 Nov 2022", "course_enroll_limit": 100, "user_id": 5)
```

INDEX (Get ALL ALREADY COURSES) —

```
curl http://localhost:3000/courses
```

— COURSES OF AN INSTRUCTOR

```
curl http://localhost:3000/instructor\_courses?id=5
```

RESPONSE::

```
{ "courses": [ { "id": 1, "name": "dsa", "subject": null, "category": null, "description": null, "start_date": "2022-11-29", "course_enroll_limit": 20, "user_id": 5, "created_at": "2022-11-30T03:34:46.549Z", "updated_at": "2022-11-30T03:34:46.549Z" }, { "id": 2, "name": "dsa", "subject": null, "category": null, "description": null, "start_date": null, "course_enroll_limit": 20, "user_id": 5, "created_at": "2022-11-30T03:35:00.877Z", "updated_at": "2022-11-30T03:35:00.877Z" }, { "id": 3, "name": "Data Structure and Algorithms", "subject": null, "category": null, "description": null, "start_date": null, "course_enroll_limit":
```

```
null,"user_id":5,"created_at":"2022-11-30T03:49:52.050Z","updated_at":"2022-11-30T03:49:52.050Z"},{"id":4,"name":"Data Structure and Algorithms","subject":"computer science","category":"beginners","description":"A complete introduction to data structures and algorithms","start_date":"2022-11-29","course_enroll_limit":100,"user_id":5,"created_at":"2022-11-30T04:01:00.152Z","updated_at":"2022-11-30T04:01:00.152Z"},{"id":5,"name":"Data Structure and Algorithms","subject":"computer science","category":"beginners","description":"A complete introduction to data structures and algorithms","start_date":"2022-11-29","course_enroll_limit":100,"user_id":5,"created_at":"2022-11-30T04:03:02.636Z","updated_at":"2022-11-30T04:03:02.636Z"},{"id":6,"name":"Data Structure and Algorithms","subject":"computer science","category":"beginners","description":"A complete introduction to data structures and algorithms","start_date":"2022-11-29","course_enroll_limit":100,"user_id":5,"created_at":"2022-11-30T04:04:35.664Z","updated_at":"2022-11-30T04:04:35.664Z"},{"id":7,"name":"Data Structure and Algorithms","subject":"computer science","category":"beginners","description":"A complete introduction to data structures and algorithms","start_date":"Tue, 29 Nov 2022","course_enroll_limit":100,"user_id":5,"created_at":"2022-11-30T04:07:00.376Z","updated_at":"2022-11-30T04:07:00.376Z"}]}
```

ENROLLMENTS :-

Create –

```
curl --header "Content-Type: application/json" --request POST --data '{"enrollment": {"course_id": 7, "user_id": 5}}' http://localhost:3000/enrollments
```

STUDENT COURSES — ENROLLED COURSES OF A STUDENT –

```
curl http://localhost:3000/student_courses?user_id=5
```

COURSE MODULE :-

CREATE A COURSE MODULE FOR A COURSE:

```
curl --header "Content-Type: application/json" --request POST --data
 '{"module": {"name": "Module 1", "position": 1, "description": "Just a
 basic desc", "media_url":
 "https://www.youtube.com/watch?v=Q3TI27IN7X0&list=RDQ3TI27IN7X0&start\_radio=1", "course_id": 7}}'
http://localhost:3000/course\_modules
```

Get modules for a course –

```
curl http://localhost:3000/course\_module\_list?course\_id=7
```

DATABASE SCHEMA:

```
create_table "course_modules", force: :cascade do |t|
  t.string "name"
  t.integer "position"
  t.text "description"
  t.bigint "course_id", null: false
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.text "media_url"
  t.index ["course_id"], name: "index_course_modules_on_course_id"
end

create_table "courses", force: :cascade do |t|
```

```

t.string "name"
t.string "subject"
t.string "category"
t.text "description"
t.string "start_date"
t.integer "course_enroll_limit"
t.bigint "user_id", null: false
t.datetime "created_at", null: false
t.datetime "updated_at", null: false
t.index ["user_id"], name: "index_courses_on_user_id"
end

create_table "enrollments", force: :cascade do |t|
  t.string "enrolled_at"
  t.integer "student_progress_module_id"
  t.bigint "course_id", null: false
  t.bigint "user_id", null: false
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.index ["course_id"], name: "index_enrollments_on_course_id"
  t.index ["user_id"], name: "index_enrollments_on_user_id"
end

create_table "users", force: :cascade do |t|
  t.string "email"
  t.string "password_digest"
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.string "role"
  t.string "dob"
  t.string "phone"
  t.string "name"
end

add_foreign_key "course_modules", "courses"
add_foreign_key "courses", "users"
add_foreign_key "enrollments", "courses"
add_foreign_key "enrollments", "users"

```

ROUTES IN THE APP:

Prefix || Verb || URI Pattern || Controller#Action

sessions	POST	/sessions(.:format)	sessions#create
registrations	GET	/registrations(.:for mat)	registrations#index
	POST	/registrations(.:for mat)	registrations#create
registration	PATCH	/registrations/:id(.: format)	registrations#update
	PUT	/registrations/:id(.: format)	registrations#update
enrollments	POST	/enrollments(.:for mat)	enrollments#create
enrollment	PATCH	/enrollments/:id(.:f ormat)	enrollments#update
	PUT	/enrollments/:id(.:f ormat)	enrollments#update

	DELETE	/enrollments/:id(.:format)	enrollments#destroy
course_modules	POST	/course_modules(.:format)	course_modules#create
course_module_list	GET	/course_module_list(.:format)	course_modules#course_module_list
student_courses	GET	/student_courses(.:format)	enrollments#student_courses
instructor_courses	GET	/instructor_courses(.:format)	courses#instructor_courses
courses	GET	/courses(.:format)	courses#index
	POST	/courses(.:format)	courses#create
Enrollment	POST	/enrollments(.:format)	enrollments#create
	GET	/student_courses(.:format)	enrollments#student_courses
logout	DELETE	/logout(.:format)	sessions#logout
logged_in	GET	/logged_in(.:format)	sessions#logged_in
root	GET	/	static#home

CROSS ORIGIN RESOURCE SHARING FUNCTIONALITY:

```
Rails.application.config.middleware.insert_before 0, Rack::Cors do
  allow do
    #change this as per the react app's local server port
    origins "http://localhost:3001"
    resource "*", headers: :any, methods: [:get, :post, :put, :patch, :delete,
:options, :head], credentials: true
  end

  allow do
    #change this as per the heroku app domain
    origins "https://silly-frangipane-f9d81d.netlify.app"
    resource "*", headers: :any, methods: [:get, :post, :put, :patch, :delete,
:options, :head], credentials: true
  end
end
```

SESSION STORE FACILITIES:

```
#correct_domain_when_domain_name_exist
if Rails.env == "production"
  Rails.application.config.session_store :cookie_store, key: "_finadvice_app",
domain: :all
else
  Rails.application.config.session_store :cookie_store, key: "_finadvice_app"
end
```

USED GEMS/LIBRARIES :-

```
source "https://rubygems.org"

git_source(:github) { |repo| "https://github.com/#{repo}.git" }

ruby "2.7.1"
```



```
# Bundle edge Rails instead: gem "rails", github: "rails/rails", branch:
"main"
gem "rails", "~> 7.0.4"

gem "pry"
# The original asset pipeline for Rails
[https://github.com/rails/sprockets-rails]
gem "sprockets-rails"

# Use postgresql as the database for Active Record

gem 'sqlite3', group: :development

gem "pg", "~> 1.1"

# gem 'pg', '0.18.1', group: :production

gem 'rails_12factor', group: :production
# Use the Puma web server [https://github.com/puma/puma]
gem "puma", "~> 5.0"

# Use JavaScript with ESM import maps
[https://github.com/rails/importmap-rails]
gem "importmap-rails"

# Hotwire's SPA-like page accelerator [https://turbo.hotwired.dev]
gem "turbo-rails"

# Hotwire's modest JavaScript framework [https://stimulus.hotwired.dev]
gem "stimulus-rails"

# Build JSON APIs with ease [https://github.com/rails/jbuilder]
gem "jbuilder"

# Use Redis adapter to run Action Cable in production
gem "redis", "~> 4.0"
```

```
# Use Kredis to get higher-level data types in Redis
[https://github.com/rails/kredis]
# gem "kredis"

# Use Active Model has_secure_password
[https://guides.rubyonrails.org/active_model_basics.html#securepassword]
# gem "bcrypt", "~> 3.1.7"

# Windows does not include zoneinfo files, so bundle the tzinfo-data gem
gem "tzinfo-data", platforms: %i[ mingw mswin x64_mingw jruby ]

# Reduces boot times through caching; required in config/boot.rb
gem "bootsnap", require: false

# Use Sass to process CSS
# gem "sassc-rails"

# Use Active Storage variants
[https://guides.rubyonrails.org/active_storage_overview.html#transforming-images]
# gem "image_processing", "~> 1.2"

group :development, :test do
  # See
  [https://guides.rubyonrails.org/debugging_rails_applications.html#debugging-with-the-debug-gem]
  gem "debug", platforms: %i[ mri mingw x64_mingw ]
end

group :development do
  # Use console on exceptions pages [https://github.com/rails/web-console]
  gem "web-console"

  # Add speed badges [https://github.com/MiniProfiler/rack-mini-profiler]
  # gem "rack-mini-profiler"

  # Speed up commands on slow machines / big apps
  [https://github.com/rails/spring]
```

```

# gem "spring"
end

group :test do
  # Use system testing
  [https://guides.rubyonrails.org/testing.html#system-testing]
  gem "capybara"
  gem "selenium-webdriver"
  gem "webdrivers"
end

gem 'bcrypt', '~> 3.1', '>= 3.1.18'
gem "rack-cors", :require => 'rack/cors'

```

CODE SAMPLE FOR SESSION MANAGEMENT:

```

class SessionsController < ApplicationController
  include CurrentUserConcern

  def create
    user = User
      .find_by(email: params["user"]["email"])
      .try(:authenticate, params["user"]["password"])

    if user
      session[:user_id] = user.id
      render json: {
        status: :created,
        logged_in: true,
        user: user
      }
    else
      render json: { status: 401, logged_in: false }
    end
  end
end

```

```
end

def logged_in
  if @current_user
    render json: {
      logged_in: true,
      user: @current_user
    }
  else
    render json: {
      logged_in: false
    }
  end
end

def logout
  reset_session
  render json: { status: 200, logged_out: true }
end
end
```